

Konstruktion eines Bluetooth-Thermometers

Jonas Otto

22. Mai 2017

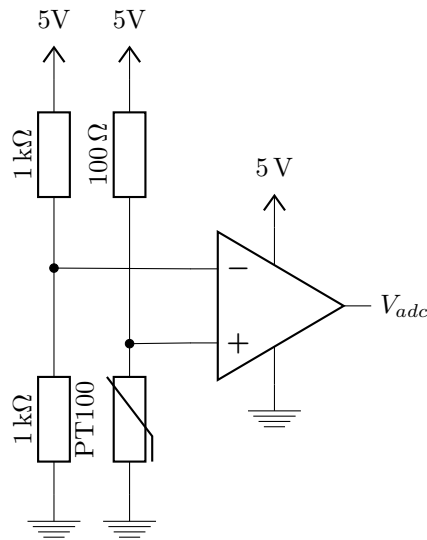
Inhaltsverzeichnis

1	Elektronik	3
1.1	Schaltplan	3
1.2	Layout	3
1.3	Bluetooth	3
2	Software	4
2.1	Microcontroller	4
2.1.1	Temperaturberechnung	5
2.1.2	Bluetooth Kommunikation	5

1 Elektronik

Verwendet wird der Temperatursensor „PT100“, der bei einer Temperatur von 0°C einen Widerstand von $100\,\Omega$ besitzt. Dieser Widerstand wird mit Hilfe eines Spannungsteilers gemessen, die Spannung V_{adc} wird mit einem Differenzverstärker verstärkt und von einem in den Arduino UNO integrierten Analog-Digital-Converter gemessen.

1.1 Schaltplan



1.2 Layout

Die Platine soll als Arduino-Shield hergestellt werden, also direkt auf das Arduino UNO Entwicklungsboard aufsteckbar sein. Dafür wurde in der Software „Target 3001!“ eine Vorlage[1] benutzt. Für eine einfache Herstellung wurde eine einseitige Platine gewählt, welche zwar für Einschränkungen im Layout sorgt, dafür aber einfach herzustellen ist. Die Bauteile befinden sich später oben, die Leiterbahnen auf der Unterseite. Es werden ausschließlich Through-Hole Komponenten verwendet, für den Operationsverstärker war ein DIL Package verfügbar, welches gesockelt verbaut wird. Die Schwierigkeit lag bei diesem Schritt darin, das Layout sowohl einfach und kompakt zu halten, als auch einen einfachen Aufbau zu gewährleisten.

1.3 Bluetooth

Für die Bluetooth Kommunikation wird ein „HC-05“ Modul eingesetzt, welches bereits mit Elektronik zur Spannungsversorgung mit 5V erhältlich ist. Da die serielle Schnittstelle des HC-05 auf einem Spannungslevel von 3.3V arbeitet, der

Arduino UNO aber auf 5 V, wurde ein Modul gewählt, welches auch Elektronik zum Level-Shifting enthält.

2 Software

Die Software besteht aus zwei Teilen: Auf dem Microcontroller wird die Temperatur berechnet und per Bluetooth an verbundene Geräte geschickt. Eine Android App zeigt die aktuelle Temperatur an und erlaubt die Konfiguration des Thermometers.

2.1 Microcontroller

Listing 1: Arduino code

```
1  #include <SoftwareSerial.h>
2
3  #define bluetoothRxBPin 10
4  #define bluetoothTxPin 11
5  #define tmpPin A0
6
7  #define cFactor -16.0 //Temperature at 0V
8  #define mFactor 0.36 //Volts per degree C
9
10 SoftwareSerial bluetoothSerial(bluetoothRxBPin, bluetoothTxPin);
11
12 void setup() {
13     bluetoothSerial.begin(9600);
14     Serial.begin(9600);
15 }
16
17 void loop() {
18     double tmpVoltage = analogRead(tmpPin) * (5.0/1023);
19     double temperatureCelsius = tmpVoltage * (1.0/mFactor) - cFactor;
20
21     Serial.print("Temperature: ");
22     Serial.print(temperatureCelsius);
23     Serial.println("C");
24
25     bluetoothSerial.println(temperatureCelsius);
26
27     delay(500);
28 }
```

2.1.1 Temperaturberechnung

Der Arduino UNO Microcontroller wird in einem angepassten C++ Dialekt programmiert. Die Hauptfunktion des Programmes ist die Umrechnung der gemessenen Spannung in eine Temperatur:

$$\Theta = V_t * \frac{1}{m} - c$$

c ist dabei die Temperatur bei 0 V, m die mittlere Steigung der Temperatur in $\frac{V}{^{\circ}C}$.

2.1.2 Bluetooth Kommunikation

Fuer die Bluetooth Kommunikation wird ein HC-05 Modul verwendet. Nach initialer Konfiguration können Daten mittels serieller Schnittstelle an das Smartphone übertragen werden. Die Temperatur wird ein Mal pro Sekunde als signed integer übertragen, gefolgt von einem newline Zeichen.

Literatur

- [1] S. Waldherr, *Target3001-templates-and-examples*, 2013 (aufgerufen 22. Mai 2017). [Online]. Verfügbar: <https://github.com/SimonWaldherr/Target3001-templates-and-examples>